Object Recognition
with Radial Basis Functions

15-496/782: Artificial Neural Networks
David S. Touretzky

Spring 2004

How Do We Recognize Objects?

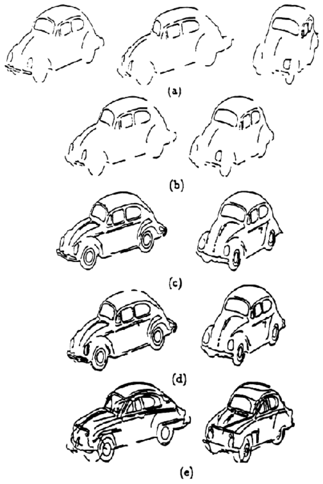
There are many sources of variation in an object's appearance:

- Observer's viewpont
- Object translation and rotation (pose)
- Configuration change (for articulated objects)
- Lighting conditions

Poggio and Colleagues: Use RBFs

Store several views of each object.

Synthesize a view-invariant recognizer from a combination of view-specific RBFs.



From Ullman & Basri (1991):

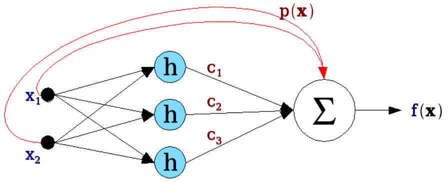
(a) Three views of VW. (b) Two synthetic views obtained by linear combination. (c) Two new views from novel viewing positions. (d) Superposition of (b) and (c) images. (e) Best fit linear combination to a different car: a Saab.

General RBF Scheme

$$f(\mathbf{x}) = \sum_{i=1}^{N} c_i \cdot h(\|\mathbf{x}-\mathbf{t}_i\|) + p(\mathbf{x})$$

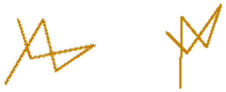
where $h(\cdot)$ is the RBF, t_i is a prototype, and c_i is its mixture coefficient (output weight)

Gaussian case: $h(\|\mathbf{x}-\mathbf{t}\|) = \exp\left(-\|\mathbf{x}-\mathbf{t}\|^2/2\sigma^2\right)$



σ small: table lookup; σ large: function interpolation

Recognizing Paperclip Shapes (Poggion & Edelman 1990)



Input: vertex positions as (x,y) coordinate pairs.

Desired output: viewpoint-invariant recognizer for that shape.

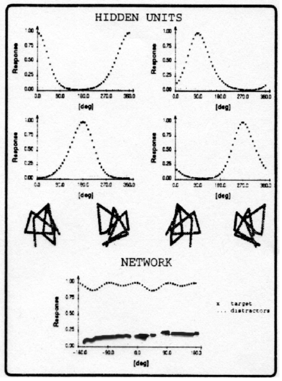
Can create a “recognition module” from several views of one object.

Responses of RBF Units in a Module

Vetter, Hurlbert, and Poggio (1995):

Four RBF units, each tuned to a different view of the object, form a “module”.

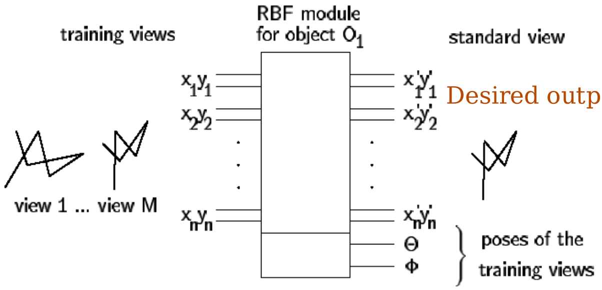
Module responds strongly to the training object, and only weakly (on average) to a set of 300 distractor objects.



Object Classification

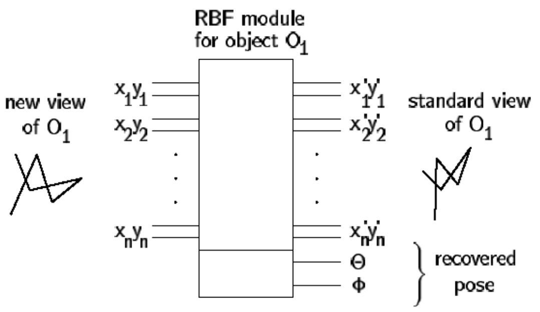
- Train each module to map an input to a “standard view”.
- Given a new image, find which module does the best job of mapping that input to the module's standard view.
- Can also recover pose information by interpolation from known posts of training instances.

Training the GRBF

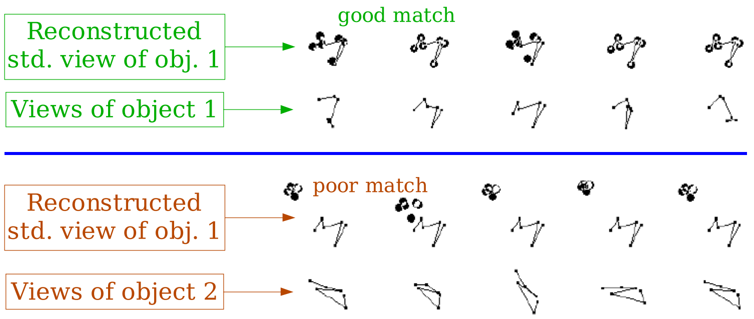


Desired output

Recovering Pose



Object Recognition


good match
Reconstructed std. view of obj. 1
Views of object 1
poor match
Reconstructed std. view of obj. 1
Views of object 2

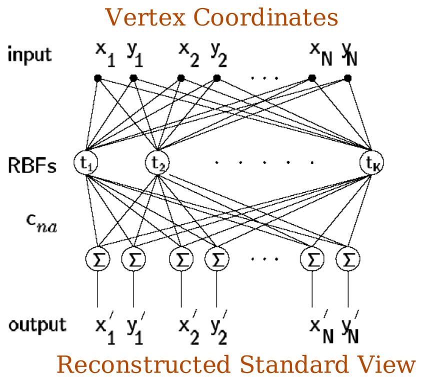
High Dimensional Input Space

Each RBF operates in a high-dimesional input space corresponding to the number of input features.

What if a feature is missing?

Example: a vertex could be hidden by occlusion.

How do we allow for a partial match?

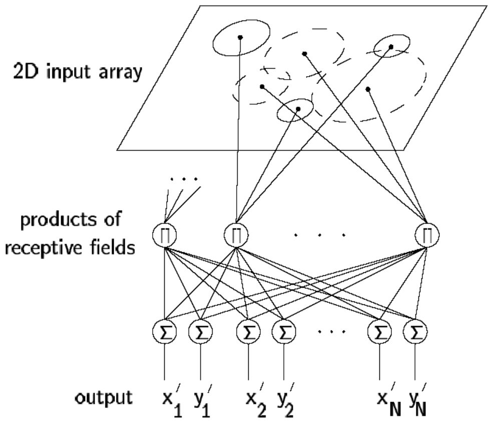

Vertex Coordinates
input
x_1 y_1 x_2 y_2 . . . x_N y_N
RBFs
t_1 t_2 t_K
c_{na}
Σ
output
x'_1 y'_1 x'_2 y'_2 x'_N y'_N
Reconstructed Standard View

Use Low-Dimensional Features

A high-dimensional Gaussian can be synthesized as a product of low-dim. Gaussians.

Each vertex can be recognized by a 2D gaussian RBF; take the product to recognize the object.

Omit occluded features.


2D input array
products of receptive fields
Π
Σ
output
x'_1 y'_1 x'_2 y'_2 x'_N y'_N

Alternative Features

RBFs are an attractive theory of how the brain might do object recognition.

But (x,y) vertices are not a biologically plausible input representation.

Alternative: could use slopes of lines, or angles between pairs of lines.

A sum of line detectors --> paperclip shape detector.

Face Detection / Object Recognition

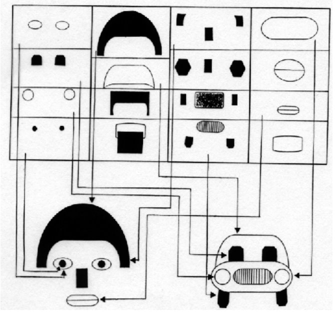
Build detectors for eyes, nose, mouth.

“Nice” objects have similar parameters under a given transformation.

Train a network to deal with this transformation.

Faces are “nice” across viewing positions.

Paperclips are not “nice”.



Hyper-Basis Functions

$$f^*(x) = \sum_{i=1}^{N} c_i \cdot G\left(\|\mathbf{x}-\mathbf{t}_i\|_W^2\right) + p(\mathbf{x})$$

$$\|\mathbf{x}-\mathbf{t}_i\|_W^2 = (\mathbf{x}-\mathbf{t}_i)^T W^T W (\mathbf{x}-\mathbf{t}_i)$$

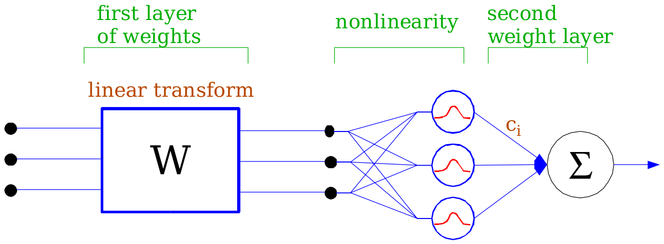
where W is a square matrix.

If W is diagonal, then the elements w_j specify weights on the input dimensions.

In the general case, W is a linear transform of the input, and the HBF acts like a multi-layer perceptron.

Train t_i, c_i, and W simultaneously.

HBF Networks



Analogy between HBG network and MLP with two layers of weights (and linear output unit.)

HBFs can sometimes learn viewpoint-invariant features, when they exist.

How to Recognize an Object From a Single Training View

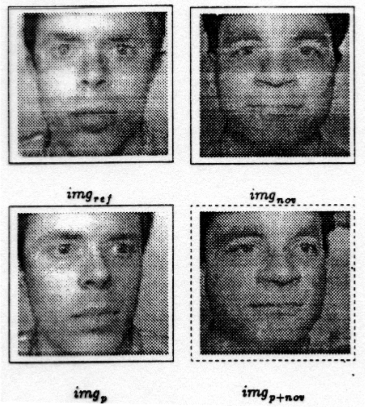
1. Develop view-invariant features by training on other, similar objects for which multiple views are available. (Training W in the HBF.)

2. Use "virtual images" generated by learned transformations to expand the training set for the new object.

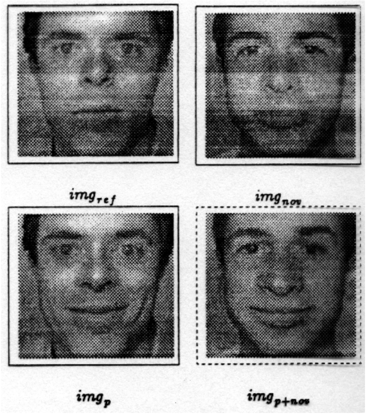
Example transformations:

- Rotation
- Change of expression (smile, frown, etc.)

Rotation Transformation



Smile Transformation



Duvdevani-Bar & Edelman: The Chorus of Prototypes Model

Shape space: description of possible 3D shapes.

Can't perceive shape directly.

"Measurement space": what is perceivable by sensors.

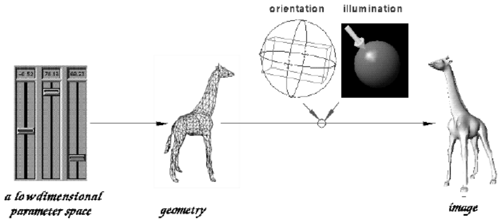
Very high dimensional space: 1 million pixels?

Rotating an object in depth generates a 2D manifold (surface) embedded in the measurement space.

The 2D manifold is the "view space" of the object.

Proximal shape space: low-dimensional parameterized description of a shape.

Image Formation



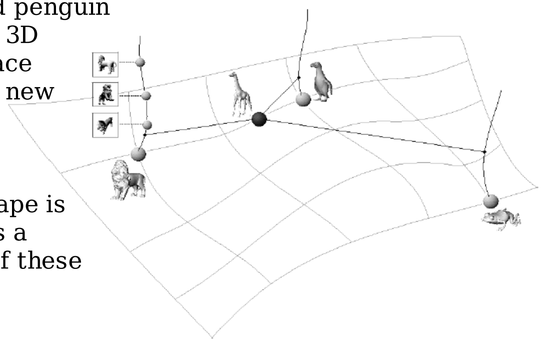
Describe the shape in a low-dimensional parameter space.

Use parameter values to generate the shape.

Apply viewpoint and lighting transforms to generate an image.

RBF Modules as Parameter Space

Lion, frog, and penguin shapes form a 3D parameter space for describing new shapes.



The giraffe shape is synthesized as a combination of these prototypes.

Chorus of Prototypes Model

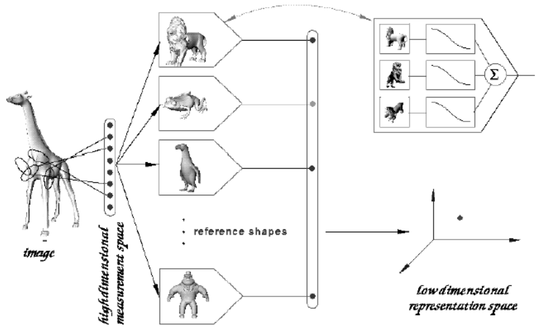
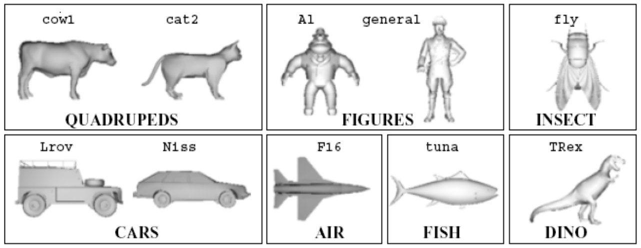


image
high dimensional measurement space
reference shapes
Σ
low dimensional representation space

Building the Model

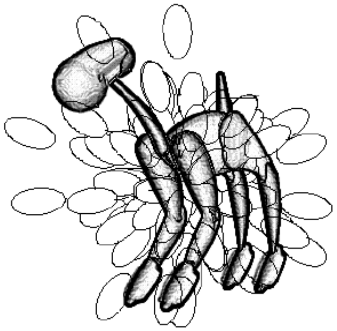
Choose 10 reference objects from a CAD database of 3D shapes.



Measurement Space

For each object, genereate a set of 169 3D views as 256x256 grayscale images.

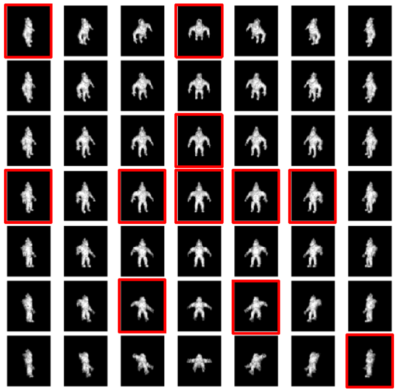
Measurement space: 200 elongated Gaussian receptive fields placed randomly over the image.



Prototype Set

Use vector quantization to cluster the 169 views of each object and generate a small number of prototypes.

This set of views defines a “module” for recognizing the object.



(The selected views are shown in red.)

Recognition Task

Identify novel views of the 10 trained objects.

Overall error rate: 7%

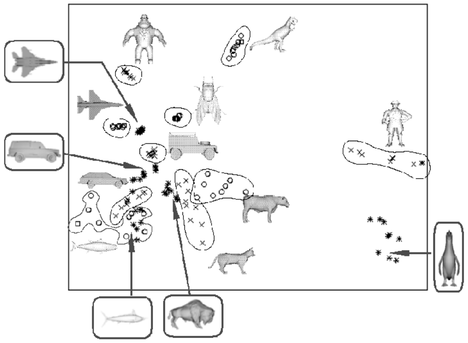
	cow1	cat	Al	gene	tuna	Lrov	Niss	F16	fly	TRex
miss rate	0.11	0.14	0.02	0.01	0.13	0.04	0.03	0.10	0.16	0.05
false alarm rate	0.08	0.11	0.07	0.02	0.11	0.05	0.04	0.12	0.12	0.03

Shape Description

Use the 10 prototypes to define a 10-dimensional space in which any shape can be described.

At right is a 2D projection of this 10D space, showing how multiple views of an object cluster together.

Cars, planes, and quadrupeds form super-clusters.

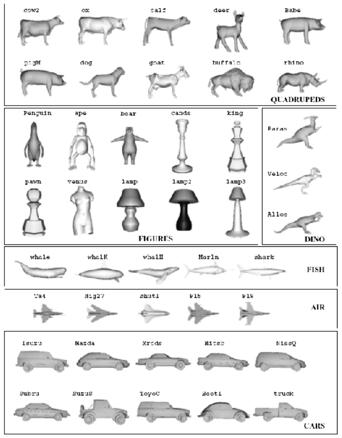


Categorization Task

Assign 43 novel objects to categories such as CARS, FISH, or QUADRUPEDS based on their locations in the 10-dimensional feature space.

Several categorization procedures were tried.

Error rate: around 20%.

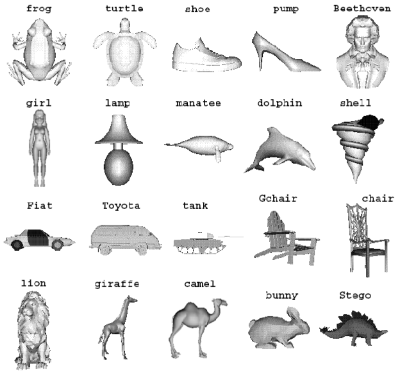


Discrimination Task

Use the 10D feature space to discriminate among 20 novel objects.

Each object is described as a point in the low-dimensional proximal shape space.

Error rate: 5%.

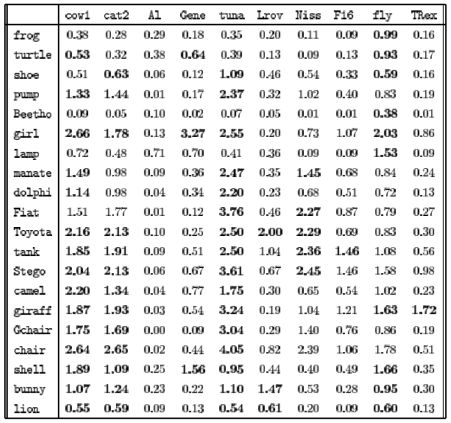


Descriptions of 20 Objects

Activations of the 10 modules for each of 20 novel objects.

Values shown in **bold** are at least 50% of the maximum activation for that row.

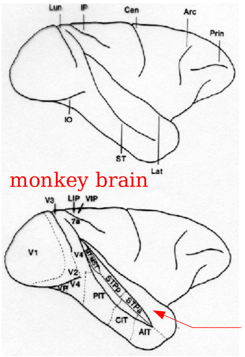
Manatee = tuna + cow + Nissan.

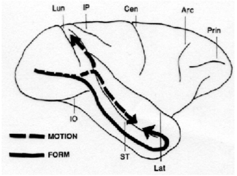


	cow1	cat2	Al	Gene	tuna	Lrov	Niss	F16	fly	TRex
frog	0.38	0.28	0.29	0.18	0.35	0.20	0.11	0.09	**0.99**	0.16
turtle	**0.53**	0.32	0.38	**0.64**	0.39	0.13	0.09	0.13	**0.93**	0.17
shoe	0.51	**0.63**	0.06	0.12	**1.09**	0.46	0.54	0.33	**0.59**	0.16
pump	**1.33**	**1.44**	0.01	0.17	**2.37**	0.32	1.02	0.40	0.83	0.19
Beetho	0.09	0.05	0.10	0.02	0.07	0.05	0.01	0.01	**0.38**	0.01
girl	**2.66**	**1.78**	0.13	**3.27**	**2.55**	0.20	0.73	1.07	**2.03**	0.86
lamp	0.72	0.48	0.71	0.70	0.41	0.36	0.09	0.09	**1.53**	0.09
manate	**1.49**	0.98	0.09	0.36	**2.47**	0.35	**1.45**	0.68	0.84	0.24
dolphi	**1.14**	0.98	0.04	0.34	**2.20**	0.23	0.68	0.51	0.72	0.13
Fiat	1.51	1.77	0.01	0.12	**3.76**	0.46	**2.27**	0.87	0.79	0.27
Toyota	**2.16**	**2.13**	0.10	0.25	**2.50**	**2.00**	**2.29**	0.69	0.83	0.30
tank	**1.85**	**1.91**	0.09	0.51	**2.50**	1.04	**2.36**	**1.46**	1.08	0.56
Stego	**2.04**	**2.13**	0.06	0.67	**3.61**	0.67	**2.45**	1.46	1.58	0.98
camel	**2.20**	**1.34**	0.04	0.77	**1.75**	0.30	0.65	0.54	1.02	0.23
giraff	**1.87**	**1.93**	0.03	0.54	**3.24**	0.19	1.04	1.21	**1.63**	**1.72**
Gchair	**1.75**	**1.69**	0.00	0.09	**3.04**	0.29	1.40	0.76	0.86	0.19
chair	**2.64**	**2.65**	0.02	0.44	**4.05**	0.82	2.39	1.06	1.78	0.51
shell	**1.89**	**1.09**	0.25	**1.56**	**0.95**	0.44	0.40	0.49	**1.66**	0.35
bunny	**1.07**	**1.24**	0.23	0.22	**1.10**	**1.47**	0.53	0.28	**0.95**	0.30
lion	**0.55**	**0.59**	0.09	0.13	**0.54**	**0.61**	0.20	0.09	**0.60**	0.13

Does the Brain Do It This Way?

There are cells in IT (inferotemporal cortex) tuned to specific shapes.





The “what” (temporal) and “where” (parietal) pathways.

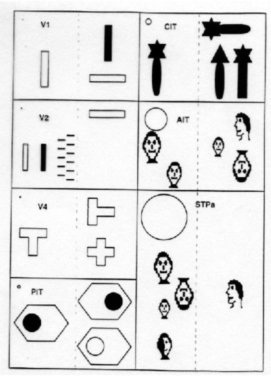
Representative Stimuli

Typical effective (left) and ineffective (right) stimuli for various visual areas.

V1 = primary visual cortex.

PIT, CIT, AIT, STPa are parts of temporal cortex.

Circles show sizes of typical receptive fields.



Responses of Cells in IT

Some cells are tuned to particular views of an object.

- Like individual prototypes in Chorus?

Some cells respond broadly to an object over a wide range of views.

- Like modules in Chorus?

Responses of cells in IT can change with experience.

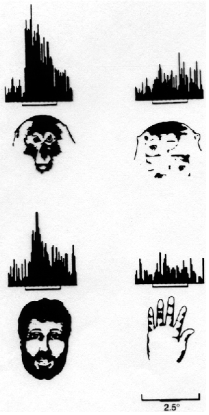
Psychophysical experiments on humans and monkeys judging shape similarity have been replicated by the Chorus of prototypes model.

Face Cells in IT

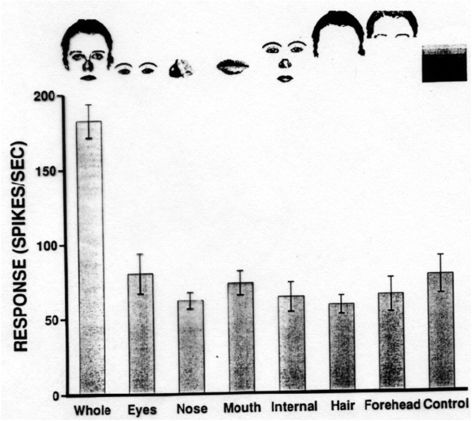
Some cells in IT are preferentially responsive to faces.

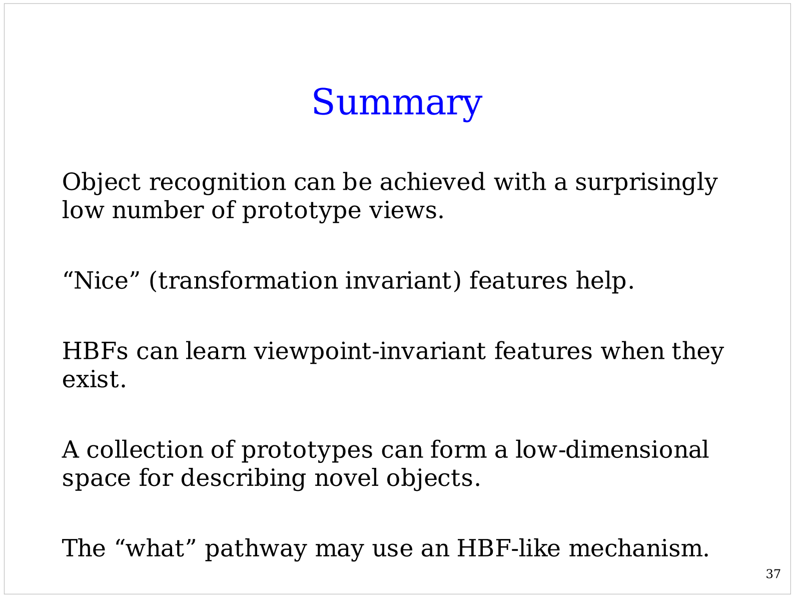
The cells will respond to either monkey faces or human faces.

They do not respond to degraded face stimuli, or non-face stimuli.



A Face Is More Than The Sum Of Its Parts





Summary

Object recognition can be achieved with a surprisingly low number of prototype views.

“Nice” (transformation invariant) features help.

HBFs can learn viewpoint-invariant features when they exist.

A collection of prototypes can form a low-dimensional space for describing novel objects.

The “what” pathway may use an HBF-like mechanism.